



HORIZONT

SmartJCL VS Code Extension

Interface for SmartJCL from Visual Studio Code

Preamble

This presentation gives an overview of SmartJCL VS Code Extension

At some slides this button can be used to jump to specific slides in this presentation.

e.g. jump to slide Agenda

Click on this symbol to jump to the overview slide

AGENDA

SmartJCL

VS Code Extension

What is SmartJCL?

JCL Validation in ISPF

JCL Validation in VS Code

Detailed Check Results View

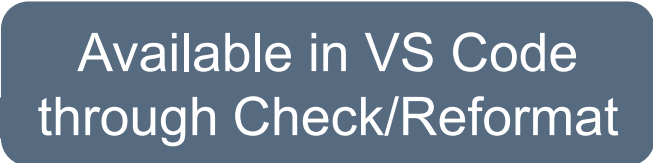
Reformat in ISPF

Reformat in VS Code

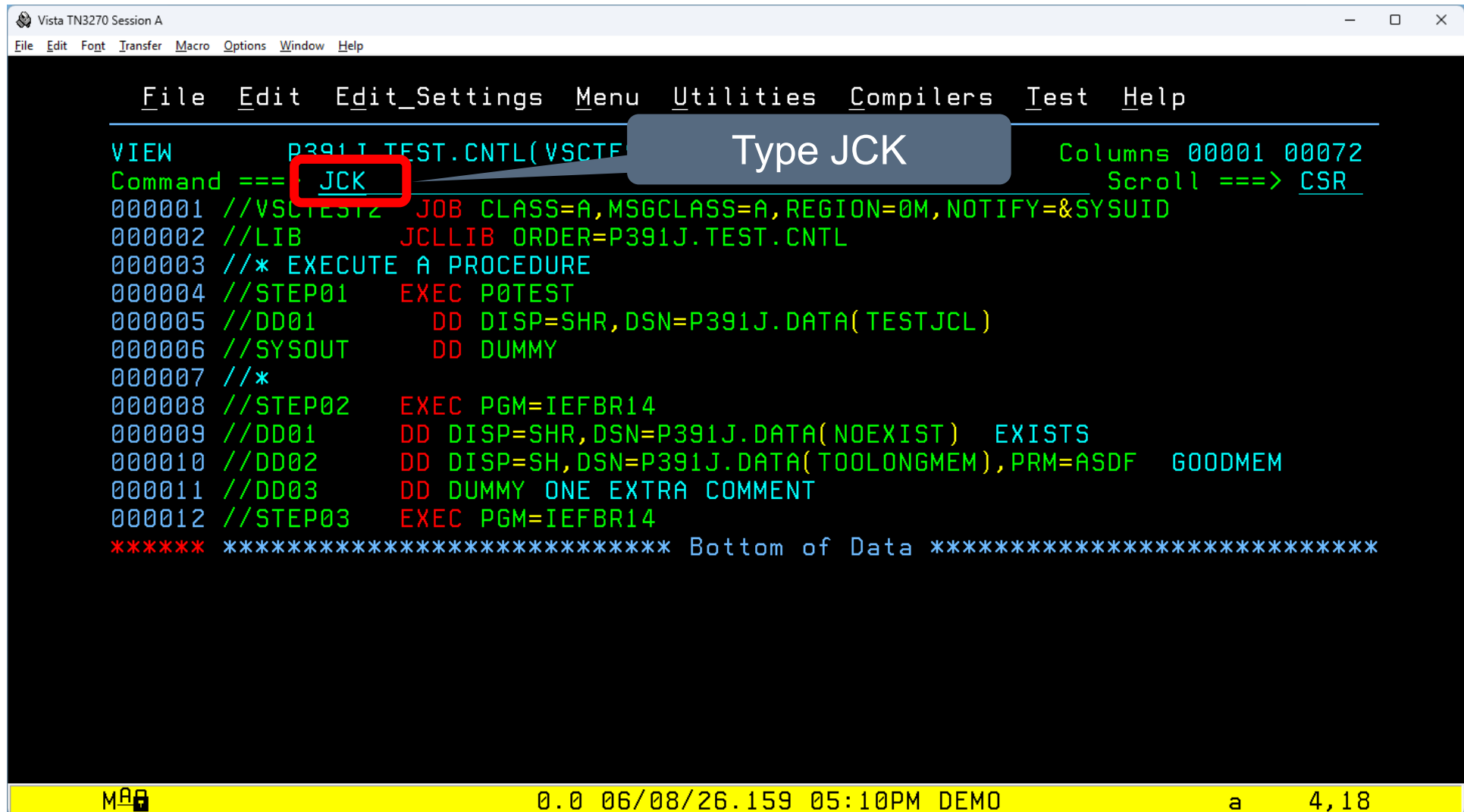
How it Works

First Time Setup

What is SmartJCL?

- It's more than a simple JCL checker. It offers many additional features that make it a complete JCL management tool:
- SmartJCL validates JCL 
 - Syntax
 - Utilities (like IDCAMS, Sort, etc.)
 - Program found / not found
 - Dataset allocation, Access authority check
- SmartJCL reformats JCL 
 - Reformat Administrator specifies positions of keyword, value and continuation line individually for each JCL parameter
 - JCL is reformatted according to these specifications
- SmartJCL Change and REXX Interface 
 - SRI provides access to all JCL statements for verification and modification by writing simple REXX code
- SmartJCL Scheduler interfaces (Control-M, IWS for z/OS)
- SmartJCL can be used as Edit Macro, in Batch, from ISPF, eclipse or from **VS Code**

JCL Validation in ISPF



Vista TN3270 Session A

File Edit Font Transfer Macro Options Window Help

File Edit Edit_Settings Menu Utilities Compilers Test Help

VIEW P391J.TEST.CNTL(VSCF) Columns 00001 00072
Command == JCK Scroll ==> CSR

```
000001 //VSCFTEST2 JOB CLASS=A,MSGCLASS=A,REGION=0M,NOTIFY=&SYSUID
000002 //LIB JCLLIB ORDER=P391J.TEST.CNTL
000003 //* EXECUTE A PROCEDURE
000004 //STEP01 EXEC P0TEST
000005 //DD01 DD DISP=SHR,DSN=P391J.DATA(TESTJCL)
000006 //SYSOUT DD DUMMY
000007 //*
000008 //STEP02 EXEC PGM=IEFBR14
000009 //DD01 DD DISP=SHR,DSN=P391J.DATA(NOEXIST) EXISTS
000010 //DD02 DD DISP=SH,DSN=P391J.DATA(TOOLONGMEM),PRM=ASDF GOODMEM
000011 //DD03 DD DUMMY ONE EXTRA COMMENT
000012 //STEP03 EXEC PGM=IEFBR14
***** ***** Bottom of Data *****
```

MR 0.0 06/08/26.159 05:10PM DEMO a 4,18

JCL Validation in ISPF

The result:

File Edit Edit_Settings Menu Utilities Help

VIEW P391J.TEST.CNTL(VSCTEST2) - 01.59 Columns 00001 00072
Command ==> Scroll ==> CSR

==MSG> .HAAC JCK0305E LENGTH OF MEMBER NAME IN DSN EXCEEDED,
==MSG> DSN=P391J.DATA(TOOLONGMEM)
==MSG> .HAAC JCK0205E UNDEFINED PARAMETER PRM ON DD STMT

000001 //VSCTEST2 JOB CLASS=A,MSGCLASS=A,REGION=0M,NOTIFY=&SYSUID
=NOTE= VSCTEST2 JOB CLASS=A,MSGCLASS=A,REGION=0M,NOTIFY=P391J
JCLLIB ORDER=P391J.TEST.CNTL
EXECUTE A PROCEDURE
P01 EXEC P0TEST
DD DISP=SHR,DSN=P391J.DATA(TESTJCL)
DD DUMMY
.HAAA /*
==MSG> JCK0080I PROCEDURE P0TEST WAS EXPANDED FROM DATA SET
==MSG> DSN=P391J.TEST.CNTL
=NOTE= P01//P0TEST PROC
=NOTE= P01//*
=NOTE= P01//* EXECUTE PROCEDURE P0TEST
=NOTE= P01//*
=NOTE= P01//P0STEP01 EXEC PGM=IEFBR14
=NOTE= P01//P0DD01 DD DISP=SHR,DSN=P391J.DATA(&PROCMEM)
==MSG> JCK0291E THE FIRST CHARACTER & OF A MEMBERNAME IS INVALID:

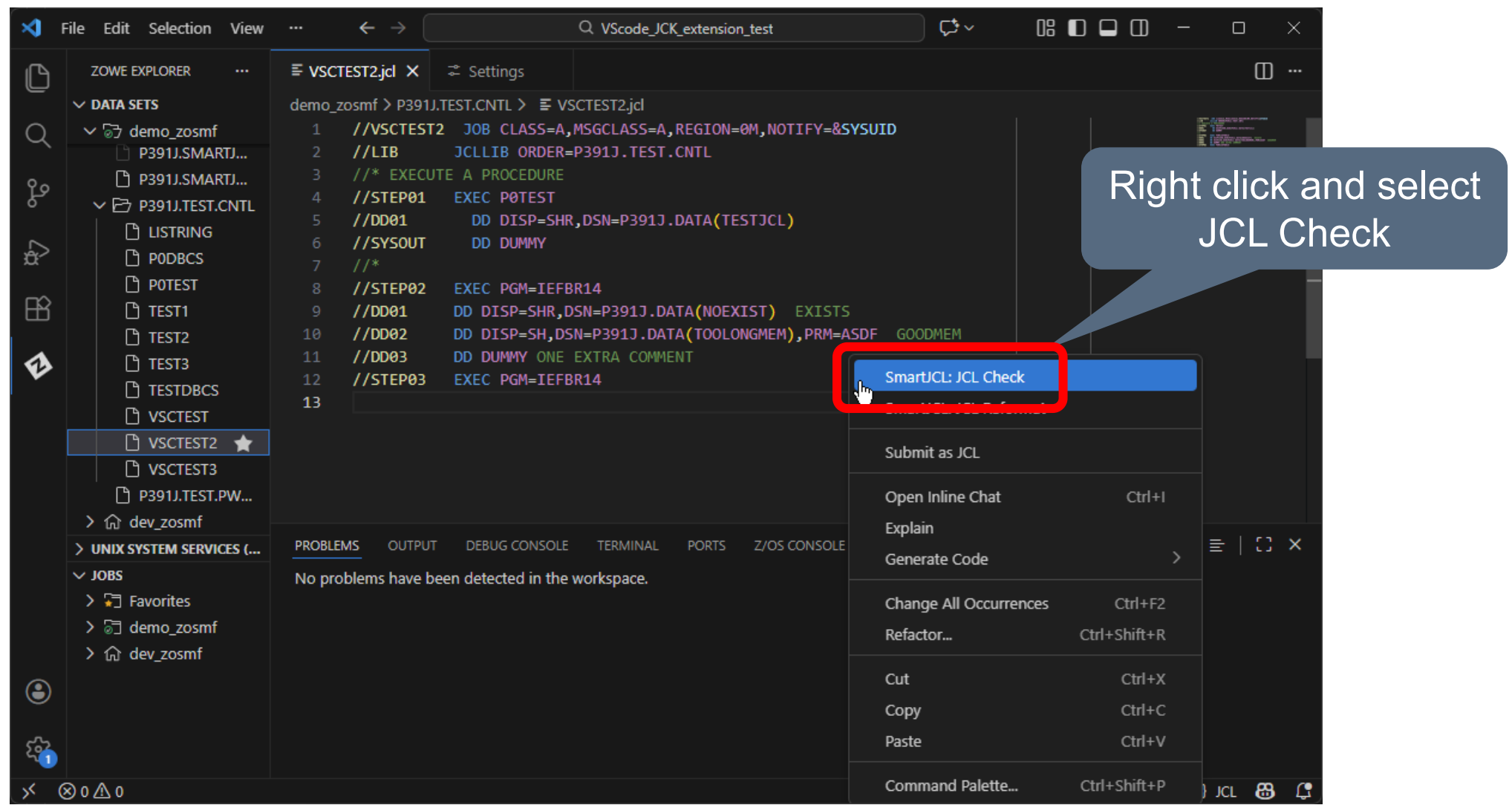
MR 0.0 06/08/26.159 05:09PM DEMO a 4,15

Labels to locate the error

Summary

Error message

JCL Validation in VS Code



JCL Validation in VS Code – Results in editor

The result:

The screenshot shows the VS Code interface with the JCL file `VSCTEST2.jcl` open. The editor displays the following JCL code:

```
1 //VSCTEST2 JOB CLASS=A,MSGCLASS=A,REGION=0M,NOTIFY=&SYSUID
2 //LIB JCLLIB ORDER=P391J.TEST.CNTL
3 /* EXECUTE A PROCEDURE
4 //STEP01 EXEC P0TEST
5 //DD01 DD DISP=SHR,DSN=P391J.DATA(TESTJCL)
6 //STEP02 EXEC PGM=IEFBR14
7 //DD01 DD DISP=SHR,DSN=P391J.DATA(NOEXIST) EXISTS
8 //DD02 DD DISP=SH,DSN=P391J.DATA(TOOLONGMEM),PRM=ASDF GOODMEM
9 //DD03 DD DUMMY ONE EXTRA COMMENT
10 //STEP03 EXEC PGM=IEFBR14
11
12
13
```

The editor underlines the following errors:

- Line 7, Col 1: `DSN=P391J.DATA(&PROCMEM)`
- Line 9, Col 1: `DSN=P391J.DATA(NOEXIST)`
- Line 10, Col 1: `DISP=SH`
- Line 10, Col 1: `DSN=P391J.DATA(TOOLONGMEM)`
- Line 10, Col 1: `PRM=ASDF`
- Line 7, Col 1: `PROCEDURE P0TEST`

The Problems panel at the bottom lists the following error messages:

- ✖ THE FIRST CHARACTER & OF A MEMBERNAME IS INVALID: DSN=P391J.DATA(&PROCMEM) JCK0291E [Ln 7, Col 1]
- ✖ MEMBER NOEXIST NOT IN DATA SET P391J.DATA JCK0318E [Ln 9, Col 1]
- ✖ INVALID VALUE SH FOR THE DISP STATUS SUBPARAMETER JCK0265E [Ln 10, Col 1]
- ✖ LENGTH OF MEMBER NAME IN DSN EXCEEDED, DSN=P391J.DATA(TOOLONGMEM) JCK0305E [Ln 10, Col 1]
- ✖ UNDEFINED PARAMETER PRM ON DD STMT JCK0205E [Ln 10, Col 1]
- ℹ PROCEDURE P0TEST WAS EXPANDED FROM DATA SET DSN=P391J.TEST.CNTL JCK0080I [Ln 7, Col 1]

Editor underlining

Summary with
message codes

JCL Validation in VS Code – Detailed results in JCKDTLO Webview

Detailed check result:

Symbol resolution

Procedure expansion

Exact inline error message location

The screenshot displays the VS Code interface with the JCL file `VSCTEST2.jcl` open. The left sidebar shows the ZOWE EXPLORER with a tree view of data sets and jobs. The main editor shows the JCL code with several annotations. A red box highlights the line `--VSCTEST2 JOB CLASS=A,MSGCLASS=A,REGION=0M,NOTIFY=&SYSUID`, with a callout indicating 'Symbol resolution'. Another red box highlights the line `JCK0080I PROCEDURE P0TEST WAS EXPANDED FROM DATA SET DSN=P391J.TEST.CNTL`, with a callout indicating 'Procedure expansion'. A third red box highlights the error message `JCK0291E THE FIRST CHARACTER & OF A MEMBERNAME IS INVALID: DSN=P391J.DATA(&PROCMEM)`, with a callout indicating 'Exact inline error message location'. The bottom panel shows the PROBLEMS view with a list of errors, including the one highlighted in the callout.

```
1 //VSCTEST2 JOB CLASS=A,MSGCLASS=A,REGION=0M,NOTIFY=&SYSUID
2 --VSCTEST2 JOB CLASS=A,MSGCLASS=A,REGION=0M,NOTIFY=P391J
3 //LIB JCLLIB ORDER=P391J.TEST.CNTL
4 /* EXECUTE A PROCEDURE
5 //STEP01 EXEC P0TEST
6 //DD01 DD DISP=SHR,DSN=P391J.DATA(TESTJCL)
7 //SYSOUT DD DUMMY
8
9 JCK0080I PROCEDURE P0TEST WAS EXPANDED FROM DATA SET DSN=P391J.TEST.CNTL
10 P01//P0TEST PROC
11 P01//*
12 P01//* EXECUTE PROCEDURE P0TEST
13 P01//*
14 P01//P0STEP01 EXEC PGM=IEFBR14
15 P01//P0DD01 DD DISP=SHR,DSN=P391J.DATA(&PROCMEM)
16 JCK0291E THE FIRST CHARACTER & OF A MEMBERNAME IS INVALID: DSN=P391J.DATA(&PROCMEM)
17 P01//
18 P01$/DD01 DD DISP=SHR,DSN=P391J.DATA(TESTJCL)
19 P01$/SYSOUT DD DUMMY
20 P01$/*
21 P01//P0STEP02 EXEC PGM=IEFBR14
```

PROBLEMS 6

- ✖ THE FIRST CHARACTER & OF A MEMBERNAME IS INVALID: DSN=P391J.DATA(&PROCMEM) JCK0291E [Ln 7, Col 1]
- ✖ MEMBER NOEXIST NOT IN DATA SET P391J.DATA JCK0318E [Ln 9, Col 1]
- ✖ INVALID VALUE SH FOR THE DISP STATUS SUBPARAMETER JCK0265E [Ln 10, Col 1]
- ✖ LENGTH OF MEMBER NAME IN DSN EXCEEDED, DSN=P391J.DATA(TOOLONGMEM) JCK0305E [Ln 10, Col 1]
- ✖ UNDEFINED PARAMETER PRM ON DD STMT JCK0205E [Ln 10, Col 1]
- ℹ PROCEDURE P0TEST WAS EXPANDED FROM DATA SET DSN=P391J.TEST.CNTL JCK0080I [Ln 7, Col 1]

JCL Reformat in ISPF

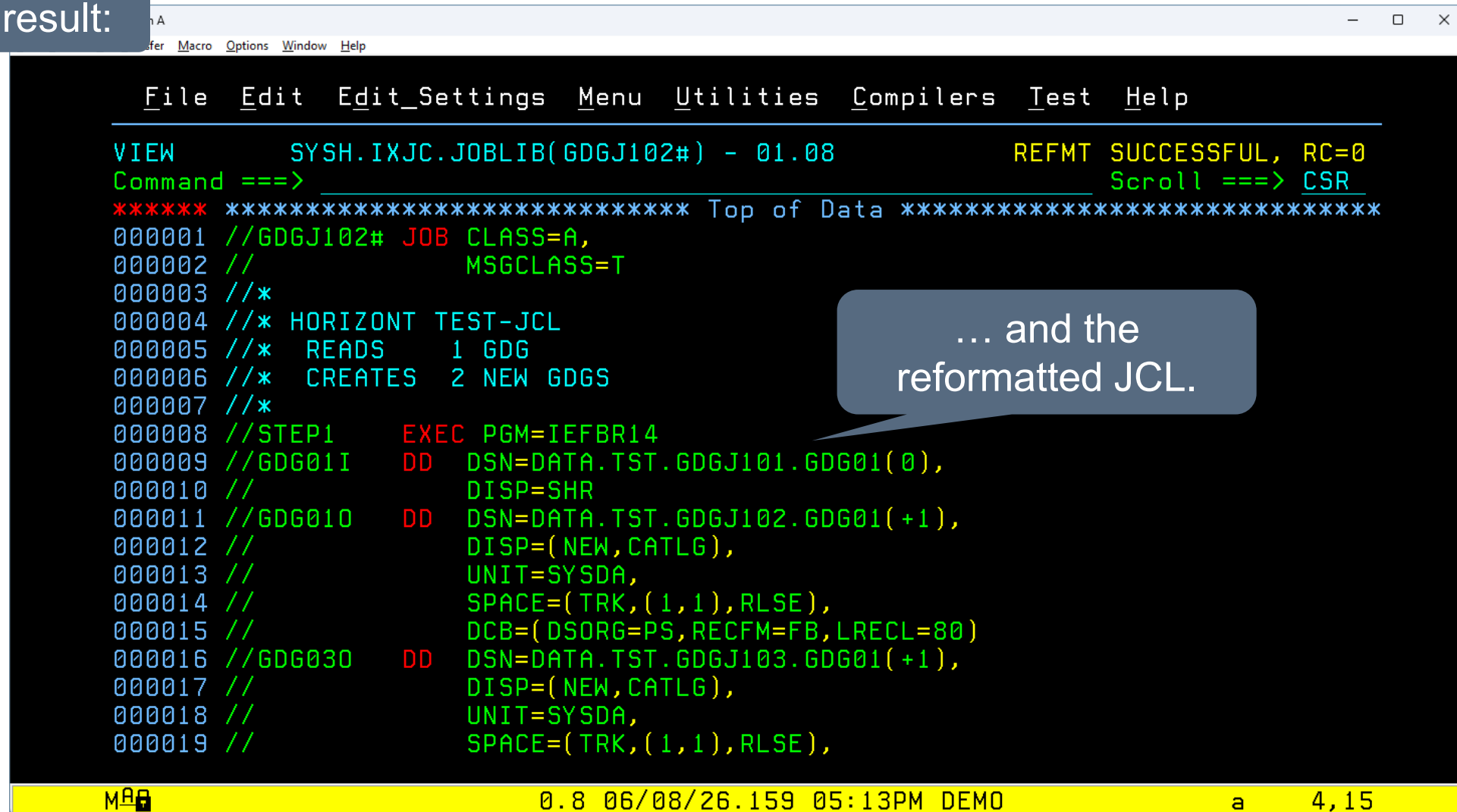
```
Vista TN3270 Session A
File Edit Edit_Settings Menu Utilities Compilers Test Help

VIEW      SYSH IX JC.JOBLIB(GDGT
Command === JREF Columns 00001 00072
Scroll ==> CSR
***** Top of Data *****
000001 //GDGJ102# JOB CLASS=A,
000002 //          MSGCLASS=T
000003 //*
000004 //* HORIZONTAL TEST-JCL
000005 //*  READS    1 GDG
000006 //*  CREATES  2 NEW GDGS
000007 //*
000008 //STEP1      EXEC PGM=IEFBR14
000009 //GDG01I     DD  DSN=DATA.TST.GDGJ101.GDG01(0),DISP=SHR
000010 //GDG010     DD  DSN=DATA.TST.GDGJ102.GDG01(+1),
000011 //  DISP=(NEW,CATLG),UNIT=SYSDA,
000012 //  SPACE=(TRK,(1,1),RLSE),
000013 //          DCB=(DSORG=PS,RECFM=FB,LRECL=80)
000014 //GDG030     DD  SPACE=(TRK,(1,1),RLSE),
000015 //  DSN=DATA.TST.GDGJ103.GDG01(+1),LRECL=80,
000016 //  DISP=(NEW,CATLG),UNIT=SYSDA,
000017 //          DCB=(DSORG=PS,RECFM=FB)
000018 //
000019 //*

MR 0.0 06/08/26.159 05:13PM DEMO a 4,19
```

JCL Reformat in ISPF

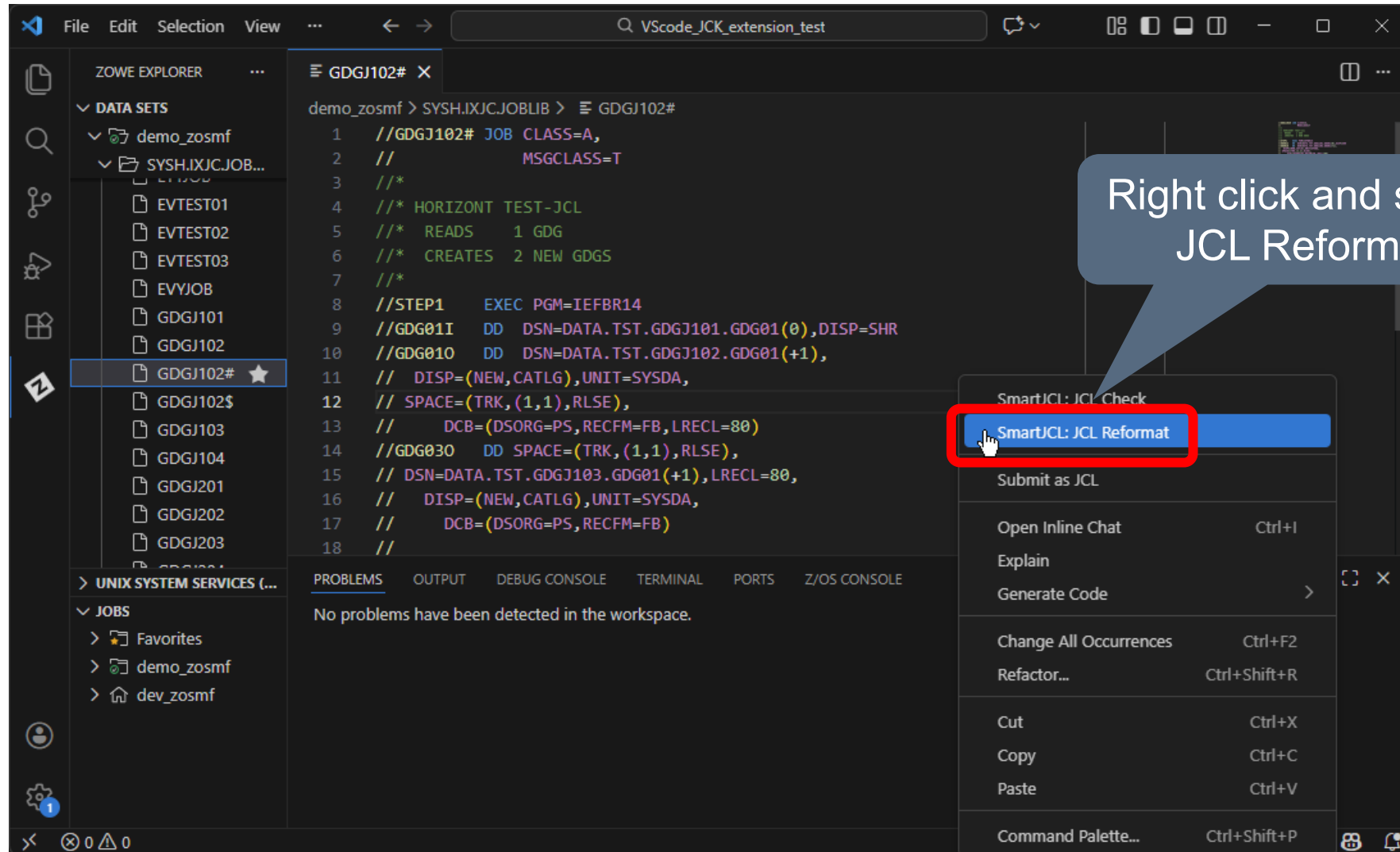
The result:



```
VIEW          SYSH.IXJC.JOBLIB(GD6J102#) - 01.08          REFMT SUCCESSFUL, RC=0
Command ==>          Scroll ==> CSR
***** Top of Data *****
000001 //GD6J102# JOB CLASS=A,
000002 //          MSGCLASS=T
000003 //*
000004 //* HORIZONT TEST-JCL
000005 //*  READS    1 GDG
000006 //*  CREATES  2 NEW GDGS
000007 //*
000008 //STEP1      EXEC PGM=IEFBR14
000009 //GD601I      DD DSN=DATA.TST.GD6J101.GD601(0),
000010 //              DISP=SHR
000011 //GD6010      DD DSN=DATA.TST.GD6J102.GD601(+1),
000012 //              DISP=(NEW,CATLG),
000013 //              UNIT=SYSDA,
000014 //              SPACE=(TRK,(1,1),RLSE),
000015 //              DCB=(DSORG=PS,RECFM=FB,LRECL=80)
000016 //GD6030      DD DSN=DATA.TST.GD6J103.GD601(+1),
000017 //              DISP=(NEW,CATLG),
000018 //              UNIT=SYSDA,
000019 //              SPACE=(TRK,(1,1),RLSE),
```

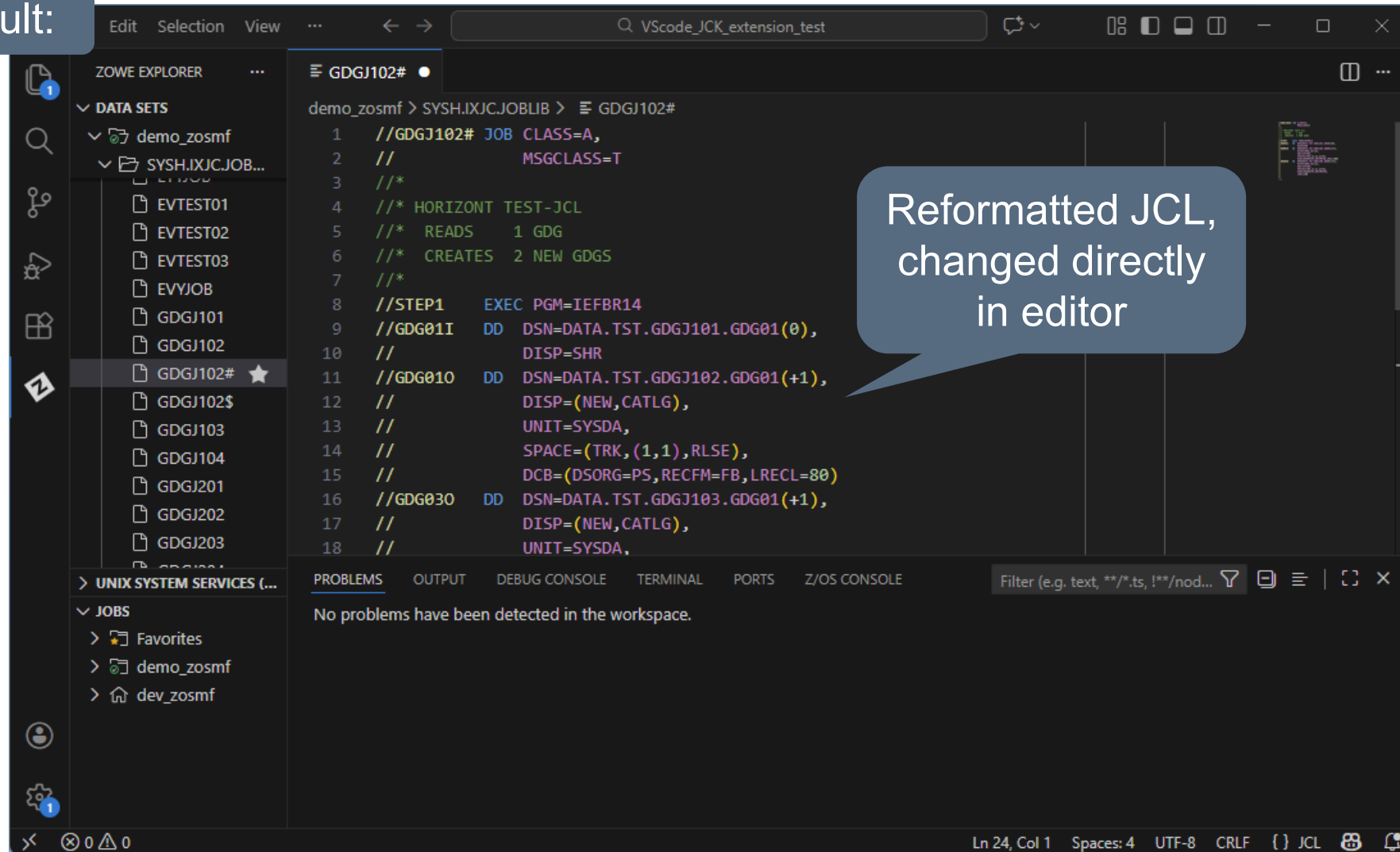
... and the reformatted JCL.

Reformat in VS Code



Reformat in VS Code

The result:



The screenshot shows the VS Code interface with the SmartJCL extension. The left sidebar displays the 'ZOWE EXPLORER' with a tree view of 'DATA SETS' and 'JOBS'. The 'demo_zosmf' folder is expanded, showing a list of files including 'GDGJ102#'. The main editor window displays the content of 'GDGJ102#', which is a JCL file. The JCL code is formatted with syntax highlighting and line numbers. A speech bubble points to the code, stating 'Reformatted JCL, changed directly in editor'. The bottom status bar indicates the file is at 'Ln 24, Col 1' with 'Spaces: 4', 'UTF-8' encoding, and 'CRLF' line endings. The 'PROBLEMS' panel at the bottom shows 'No problems have been detected in the workspace.'

```
1 //GDGJ102# JOB CLASS=A,
2 //          MSGCLASS=T
3 //
4 /* HORIZONTAL TEST-JCL
5 /* READS   1 GDG
6 /* CREATES 2 NEW GDGS
7 /*
8 //STEP1    EXEC PGM=IEFBR14
9 //GDG01I   DD DSN=DATA.TST.GDGJ101.GDG01(0),
10 //          DISP=SHR
11 //GDG010   DD DSN=DATA.TST.GDGJ102.GDG01(+1),
12 //          DISP=(NEW,CATLG),
13 //          UNIT=SYSDA,
14 //          SPACE=(TRK,(1,1),RLSE),
15 //          DCB=(DSORG=PS,RECFM=FB,LRECL=80)
16 //GDG030   DD DSN=DATA.TST.GDGJ103.GDG01(+1),
17 //          DISP=(NEW,CATLG),
18 //          UNIT=SYSDA,
```

Reformatted JCL,
changed directly
in editor

How it Works

SmartJCL VS Code Extension:

- Communicates with the host using z/OSMF
 - Zowe SDK (Software Development Kit) is used for z/OS communication via z/OSMF
- Is meant to be used with Zowe
 - It uses a Zowe z/OSMF profile to establish the z/OS session
 - Zowe Explorer (another VS Code extension) can be used to open JCL from the host
- Requires a SmartJCL installation on the z/OS
 - SmartJCL is used to process the JCL and send results back to VS Code

First Time Setup

If Zowe is already set up and your VS Code can open files in “JCL” language mode, getting the SmartJCL extension to work is easy:

1. In VS Code, download the SmartJCL extension from the marketplace

2. Open the extension settings and fill in:

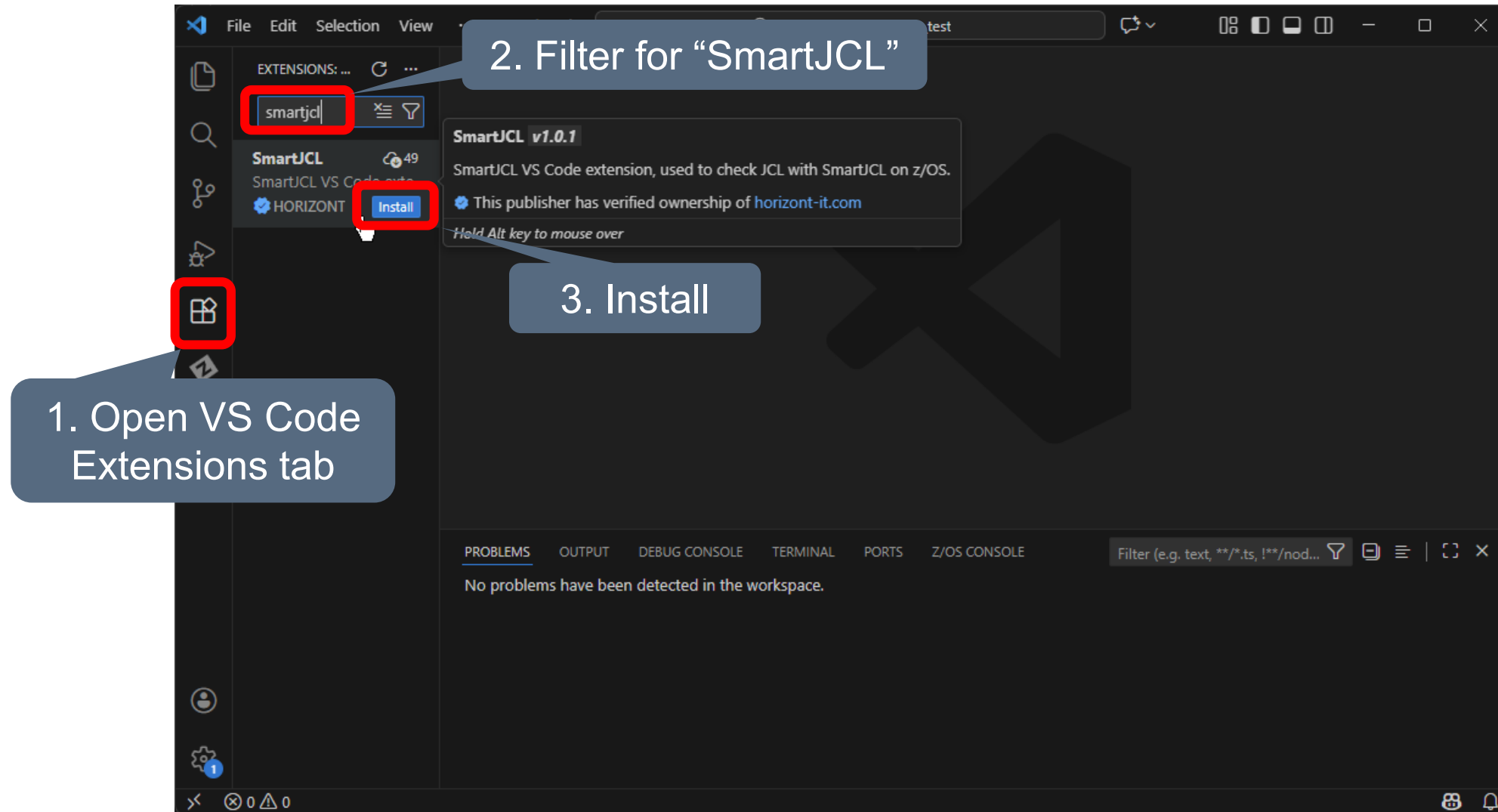
- Name of the Zowe profile to be used for z/OS connection
- Data set name of the SmartJCL parameter member

Step 2. may even be skipped:

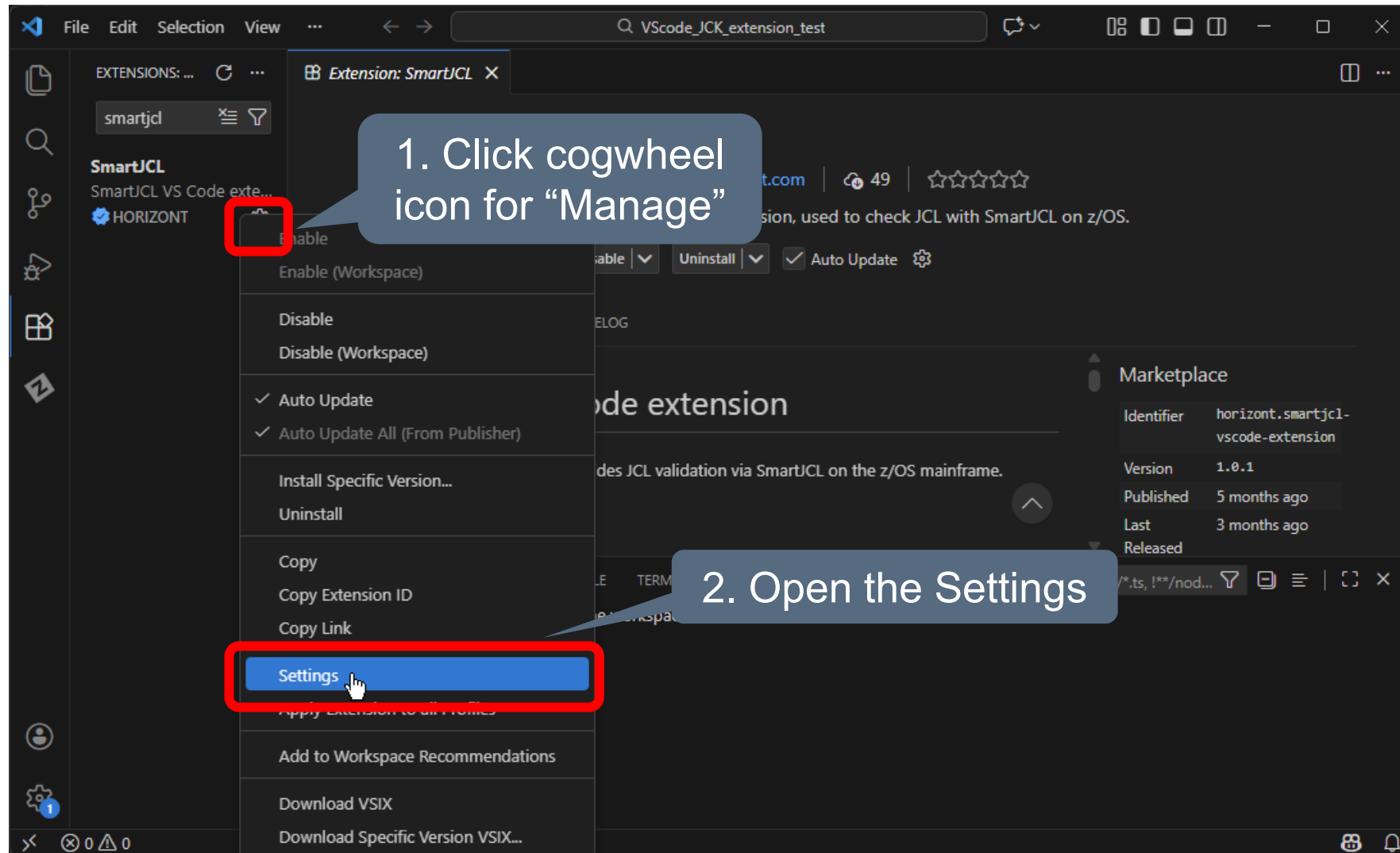
- Default Zowe z/OSMF profile will be used.
- You will be prompted to enter the JCKXPARM DSN when needed.

3. Run the command “SmartJCL: JCL Check” on your opened JCL

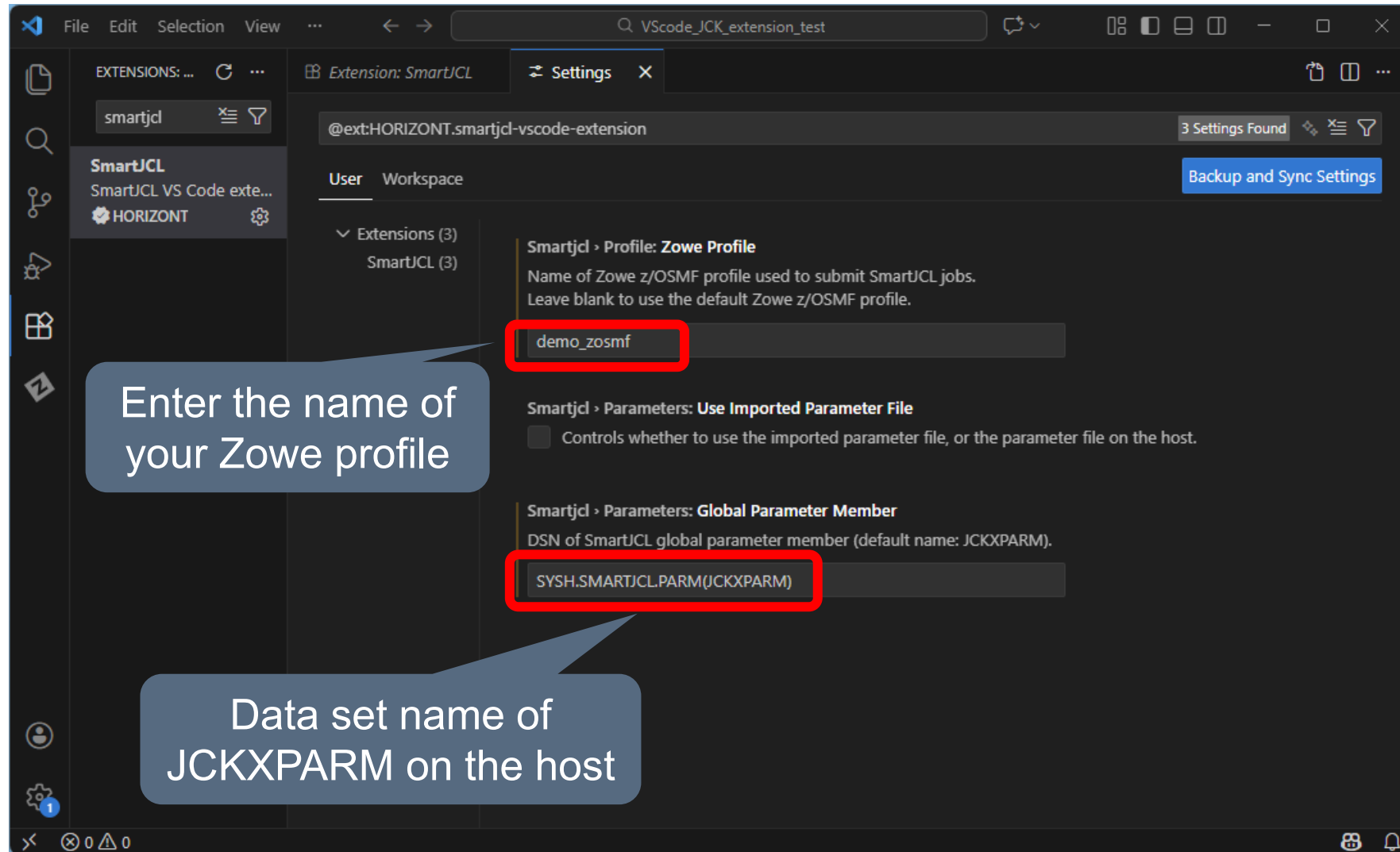
First Time Setup – 1. Download the Extension



First Time Setup – 2. Open Extension Settings



First Time Setup – 2. Edit the Settings



First Time Setup – 3. Run JCL Check

The screenshot displays the VS Code interface with the SmartJCL extension. The left sidebar shows the 'ZOWE EXPLORER' with a tree view of data sets. The main editor shows a JCL file named 'VSCTEST2.jcl'. A context menu is open over the file, highlighting the 'SmartJCL: JCL Check' option.

1. Open Zowe Explorer

2. List data sets

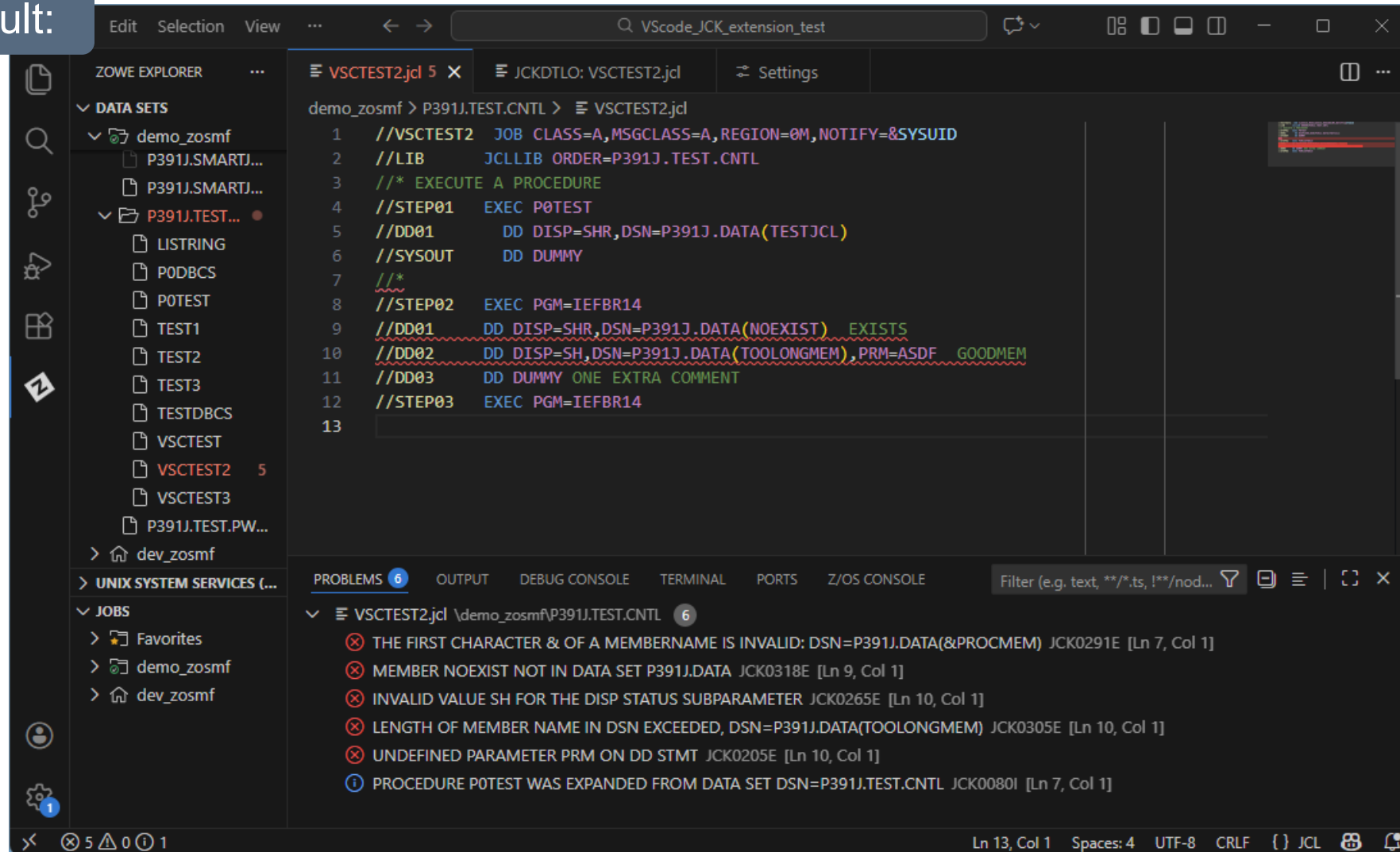
3. Open JCL member

4. Right click and select JCL Check

```
//VSCTEST2 JOB CLASS=A,MSGCLASS=A,REGION=0M,NOTIFY=&SYSUID
//DD01 DD DISP=SHR,DSN=P391J.DATA(TESTJCL)
//SYSOUT DD DUMMY
//STEP02 EXEC PGM=IEFBR14
//DD01 DD DISP=SHR,DSN=P391J.DATA(NOEXIST) EXISTS
//DD02 DD DISP=SH,DSN=P391J.DATA(TOOLONGMEM),PRM=ASDF GOODMEM
//DD03 DD DUMMY ONE EXTRA COMMENT
//STEP03 EXEC PGM=IEFBR14
```

First Time Setup – JCL Check Results

The result:



The screenshot displays the VS Code interface with the SmartJCL extension. The left sidebar shows the 'ZOWE EXPLORER' with a tree view of 'DATA SETS' and 'JOBS'. The main editor area shows the file 'VSCTEST2.jcl' with the following JCL code:

```
1 //VSCTEST2 JOB CLASS=A,MSGCLASS=A,REGION=0M,NOTIFY=&SYSUID
2 //LIB JCLLIB ORDER=P391J.TEST.CNTL
3 /* EXECUTE A PROCEDURE
4 //STEP01 EXEC P0TEST
5 //DD01 DD DISP=SHR,DSN=P391J.DATA(TESTJCL)
6 //SYSOUT DD DUMMY
7 /*
8 //STEP02 EXEC PGM=IEFBR14
9 //DD01 DD DISP=SHR,DSN=P391J.DATA(NOEXIST) EXISTS
10 //DD02 DD DISP=SH,DSN=P391J.DATA(TOOLONGMEM),PRM=ASDF GOODMEM
11 //DD03 DD DUMMY ONE EXTRA COMMENT
12 //STEP03 EXEC PGM=IEFBR14
13
```

The bottom panel shows the 'PROBLEMS' view with 6 error messages:

- ✖ THE FIRST CHARACTER & OF A MEMBERNAME IS INVALID: DSN=P391J.DATA(&PROCMEM) JCK0291E [Ln 7, Col 1]
- ✖ MEMBER NOEXIST NOT IN DATA SET P391J.DATA JCK0318E [Ln 9, Col 1]
- ✖ INVALID VALUE SH FOR THE DISP STATUS SUBPARAMETER JCK0265E [Ln 10, Col 1]
- ✖ LENGTH OF MEMBER NAME IN DSN EXCEEDED, DSN=P391J.DATA(TOOLONGMEM) JCK0305E [Ln 10, Col 1]
- ✖ UNDEFINED PARAMETER PRM ON DD STMT JCK0205E [Ln 10, Col 1]
- ℹ PROCEDURE P0TEST WAS EXPANDED FROM DATA SET DSN=P391J.TEST.CNTL JCK0080I [Ln 7, Col 1]

The status bar at the bottom indicates 'Ln 13, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'JCL'.

First Time Setup – JCL Check Detailed Results

1. Filter for “SmartJCL” commands in VS Code’s command palette

2. Open JCKDTLO Webview

```
>smartjcl
SmartJCL: Open JCKDTLO Webview
SmartJCL: Clear Check results
SmartJCL: JCL Check
SmartJCL: Import host configuration
SmartJCL: Tailor and open job
SmartJCL: JCL Reformat
```

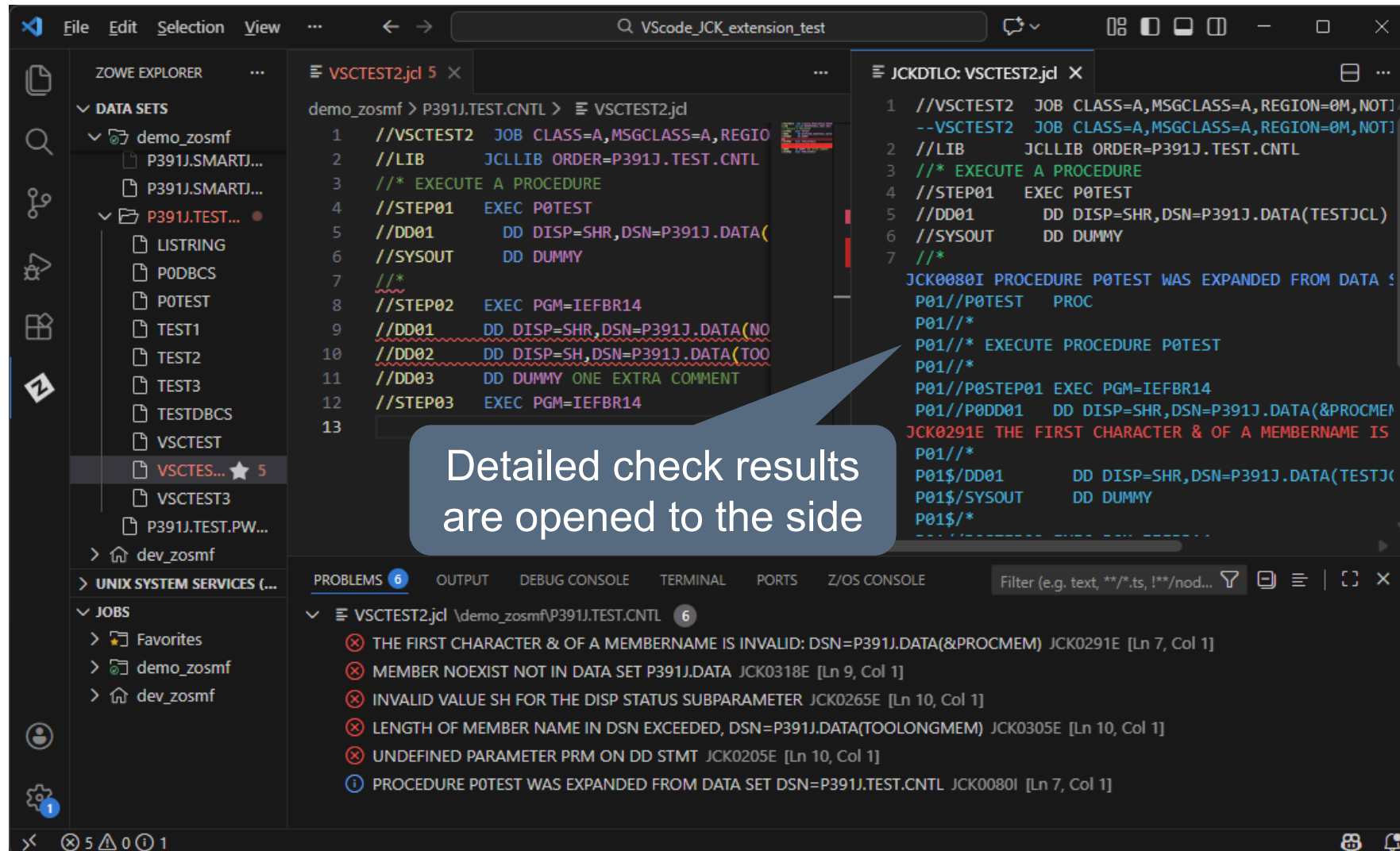
```
6 //SYSOUT DD DUMMY
7 /**
8 //STEP02 EXEC PGM=IEFBR14
9 //DD01 DD DISP=SHR,DSN=P391J.DATA(NOEXIST) EXISTS
10 //DD02 DD DISP=SH,DSN=P391J.DATA(TOOLONGMEM),PRM=ASDF GOODMEM
11 //DD03 DD DUMMY ONE EXTRA COMMENT
12 //STEP03 EXEC PGM=IEFBR14
13
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS Z/OS CONSOLE

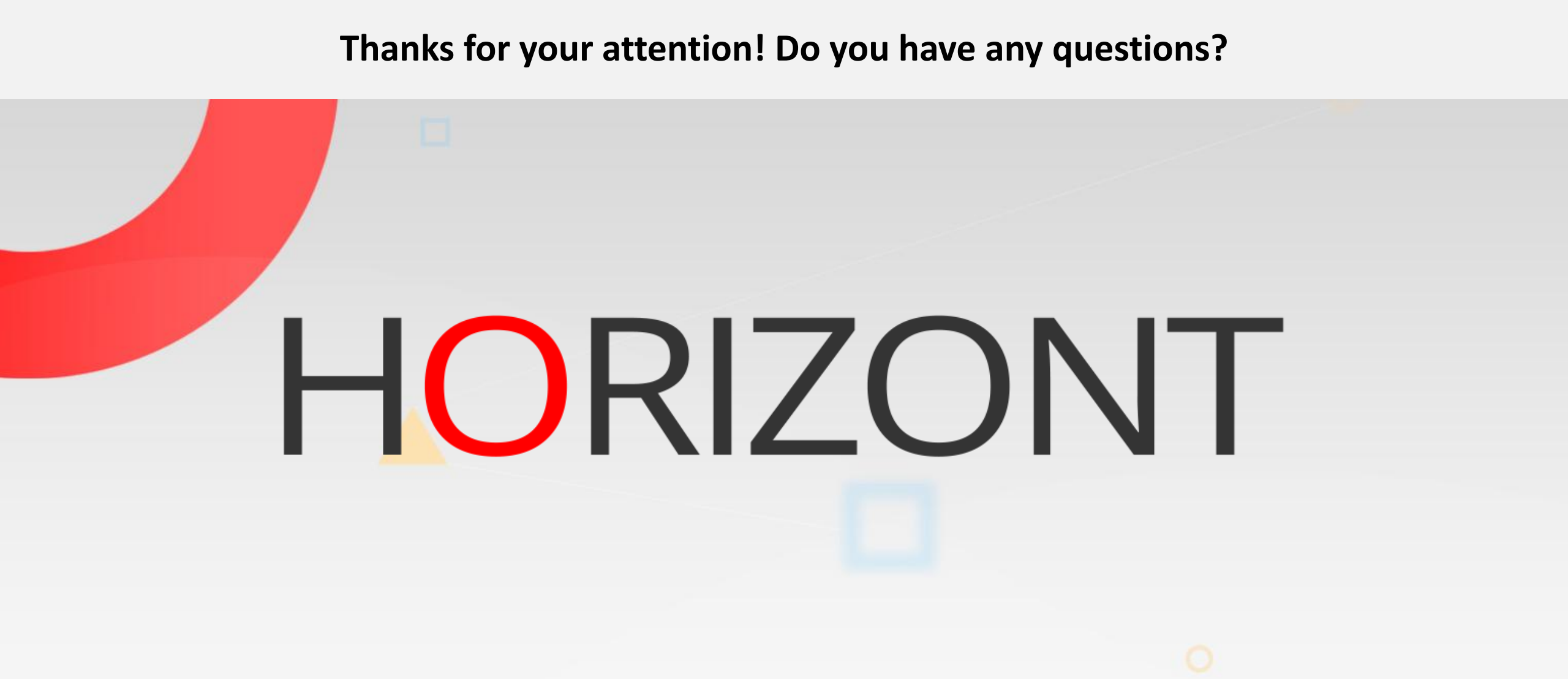
- ✗ THE FIRST CHARACTER & OF A MEMBERNAME IS INVALID: DSN=P391J.DATA(&PROCMEM) JCK0291E [Ln 7, Col 1]
- ✗ MEMBER NOEXIST NOT IN DATA SET P391J.DATA JCK0318E [Ln 9, Col 1]
- ✗ INVALID VALUE SH FOR THE DISP STATUS SUBPARAMETER JCK0265E [Ln 10, Col 1]
- ✗ LENGTH OF MEMBER NAME IN DSN EXCEEDED, DSN=P391J.DATA(TOOLONGMEM) JCK0305E [Ln 10, Col 1]
- ✗ UNDEFINED PARAMETER PRM ON DD STMT JCK0205E [Ln 10, Col 1]
- ℹ PROCEDURE P0TEST WAS EXPANDED FROM DATA SET DSN=P391J.TEST.CNTL JCK0080I [Ln 7, Col 1]

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} JCL

First Time Setup – JCL Check Detailed Results



Thanks for your attention! Do you have any questions?

The logo for 'HORIZONT' is displayed in a large, sans-serif font. The letter 'O' is a solid red circle, while the other letters are dark grey. To the left of the 'O' is a large red curved shape. Scattered around the text are several small, faint geometric shapes: a blue square, a yellow triangle, a blue square, and a yellow circle.

HORIZONT

Please feel free to visit us in Munich or send an email to
info@horizont-it.com